PURDUE UNIVERSITY
REQUEST FOR ADDITION, EXPIRATION,
OR REVISION OF AN UNDERGRADUATE COURSE
(10000-40000 LEVEL)

DEPARTMENT: COMPUTER SCIENCE
EFFECTIVE SESSION: FALL 2011

INSTRUCTIONS: Please check the items below which describe the purpose of this request.

☐ 1. New course with supporting documents
☐ 2. Add existing course offered at another campus
☐ 3. Expiration of a course
☐ 4. Change in course number
☐ 5. Change in course title
☐ 6. Change in course credit type
☐ 7. Change in course attributes (department head signature only)
☐ 8. Change in instructional hours
☐ 9. Change in course description
☐ 10. Change in course requisites
☐ 11. Change in semester offered (department head signature only)
☐ 12. Transfer from one department to another

PROPOSED:

Subject Abbreviation: CS
Course Number: 46000
Long Title: SENIOR CAPSTONE PROJECT I.
Short Title: See Proj I

EXISTING:

Subject Abbreviation: CS
Course Number: 46000

TERMS OFFERED:
Check All That Apply:
☐ Fall ☐ Spring ☐ Summer

CAMPUS(ES) INVOLVED:

☐ Calumet
☐ Cont Ed
☐ Ft. Wayne
☐ Tech Statewide
☐ Indianapolis
☐ W. Lafayette

Abbreviated title will be entered by the Office of the Registrar if omitted. (50 CHARACTERS ONLY)

CREDIT TYPE:

1. Fixed Credit: Cr. Hrs.
   ☐ 3
2. Variable Credit Range:
   ☐ Minimum Cr. Hrs
   ☐ (Check One) To ☐ Or ☐
   ☐ Maximum Cr. Hrs.
3. Equivalent Credit: Yes ☐ No ☐

COURSE ATTRIBUTES:
Check All That Apply:

1. Pass/Not Pass Only
2. Satisfactory/Unsatisfactory Only
3. Repeatable
4. Maximum Repeated Credit:
5. Credit by Examination
6. Registration Approval Type
   ☐ Department ☐ Instructor
7. Variable Title
8. Honors
9. Full Time Privilege
10. Off Campus Experience

SCHEDULE/TYP:

Lecture
Recitation
Presentation
Laboratory
Lab Prep
Studio
Distance
Clinic
Experiential
Research
Ind. Study

Meetings Per Week

Weeks Offered

% of Credit
Allocated

Cross-Listed Courses

COURSE DESCRIPTION (INCLUDE REQUISITES/RESTRICTIONS):
P: CS 36000 and Senior Standing. The first course of a two-semester sequence. Student teams will participate in the development of a substantial application-oriented or research-oriented software project utilizing a formal software process model. Emphasis on teamwork, project management, and oral and written communication. Student teams will conduct review activities and develop artifacts appropriate for the software project and process model chosen.

COURSE LEARNING OUTCOMES:
1. Apply software engineering principles and skill to a team-oriented software project
2. Construct a software project schedule and track its progress
3. Construct artifacts appropriate to demonstrate completion of each phase of software process
4. Conduct formal technical reviews
5. Utilize a repository for project artifacts

Column Department Head
Date

Fort Wayne Department Head
Date

Indianapolis Department Head
Date

North Central Faculty Senate Chair
Date

West Lafayette Department Head
Date

Counselor School Dean
Date

Fort Wayne School Dean
Date

Indianapolis School Dean
Date

Vice Chancellor for Academic Affairs
Date

West Lafayette College/School Dean
Date

West Lafayette Registrar
Date

OFFICE OF THE REGISTRAR
<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>CS 46000</th>
<th>Course Title</th>
<th>Senior Capstone Project 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Robert Sedlmeyer</td>
</tr>
</tbody>
</table>

Current Catalog Description

P: CS 36000 and Senior Standing. The first course of a two-semester sequence. Student teams will participate in the development of a substantial application-oriented or research-oriented software project utilizing a formal software process model. Emphasis on teamwork, project management, and oral and written communication. Student teams will conduct review activities and develop artifacts appropriate for the software project and process model chosen.

Textbook

1. None.

References

1. None.

Course Outcomes

1. Apply software engineering principles and skills to a team-oriented software project (a, b, c, d, i, k)
2. Construct a software project schedule and track its progress (a, i, k)
3. Construct artifacts appropriate to demonstrate completion of each phase of the software process. (a, i, k)
4. Conduct formal technical reviews (f, k)
5. Utilize a repository for project artifacts (a, i, k)

Relationship between Course Outcomes and Program Outcomes

The numbered Course Outcomes support the Program Outcomes as indicated in the following table, where the Program Outcomes (a-k) are listed below the table:
<table>
<thead>
<tr>
<th>Course Outcome</th>
<th>Program Outcomes</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

**Program Outcomes**

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
d. An ability to function effectively on teams to accomplish a common goal.
e. An understanding of professional, ethical, legal, security and social issues and responsibilities.
f. An ability to communicate effectively with a range of audiences.
g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
h. Recognition of the need for and an ability to engage in continuing professional development.
i. An ability to use current techniques, skills, and tools necessary for computing practice.
j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
k. An ability to apply design and development principles in the construction of software systems of varying complexity.

**Prerequisites by Topic**

1. Principles of Software Engineering
2. Ethics and Professionalism
3. Software Process Models
4. Unified Modeling Language
5. Team-oriented Software Development
6. Software Requirements and Specification
7. Architecture and Software Design
8. Software Development
9. Software Validation and Verification
10. Software Project Management
Major Topics Covered in the Course

No additional topics will be covered.

Assessment Plan for the Course

The course utilizes two tools for assessment: Student performance on the first part of the project, which may be solicited from an external client, and the results of a student survey. The project is evaluated by the project director (a Department faculty member) and other stakeholders (e.g., client, Departmental faculty). This provides an objective measure of level-of-achievement for course outcomes. The survey asks students to judge how well they achieved each learning outcome on a 5-pt Likert scale. The instructor is given the results and encouraged to make recommendations for the next offering, especially for those course outcomes whose average objective or subjective scores fall below 3.

How Data in the Course is Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

The results of the instructor and client project evaluation and student survey are the principal means by which program outcomes are assessed.

Estimate Curriculum Category Content (Semester hours)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td></td>
<td></td>
<td>Software design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data structures</td>
<td></td>
<td></td>
<td>Concepts of programming</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>languages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer organization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>